

INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
BAHIA  
Campus Salvador

# Escalonamento de Processos

---

---

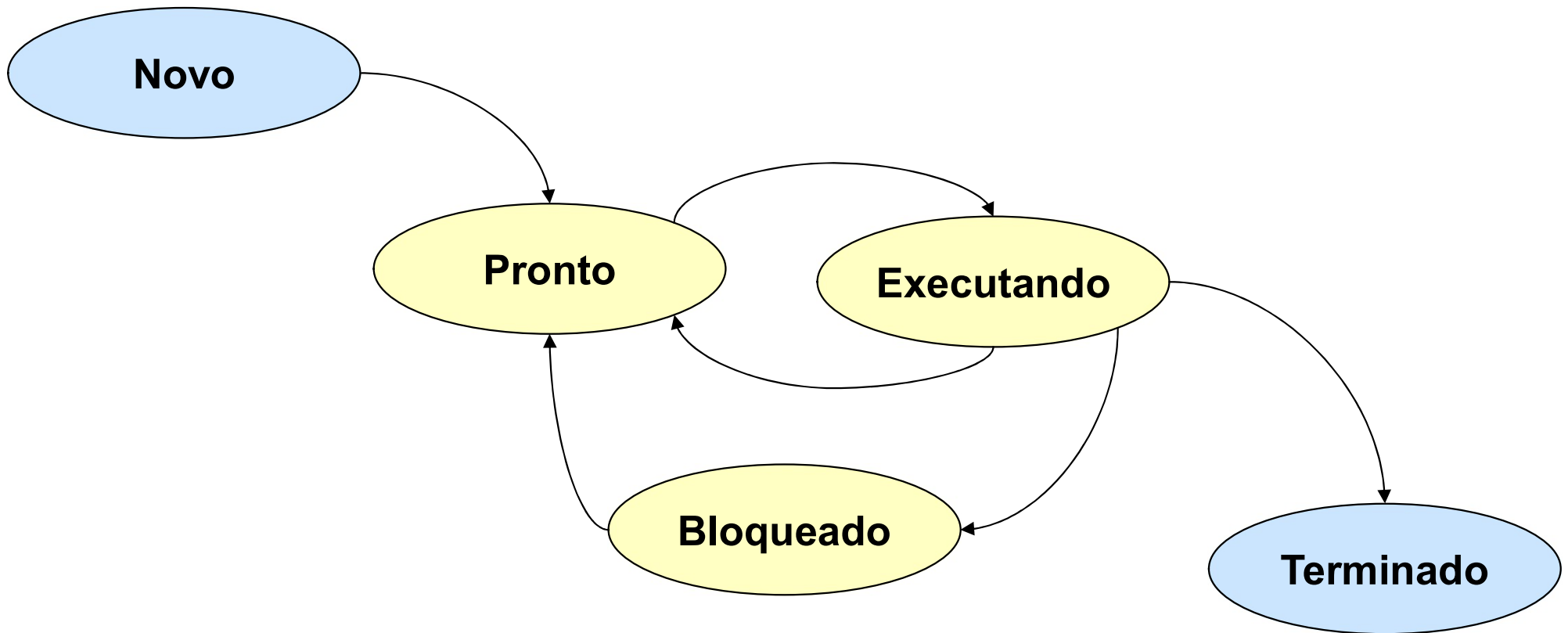
# Porque é necessário escalonar?

---

- Processos precisam ser executados
- Processos concorrem a CPU
- Escalonador:
  - Componente (implementação) do sistema operacional
  - Determina a ordem de execução dos processos baseado num *algoritmo de escalonamento*
  - Lê a fila que contém os processos no estado “pronto” e os ordena para execução

# O que provoca o escalonamento?

---



# Tipos de algoritmo de escalonamento

---

## ■ Preemptivo:

- Execução de um processo dura um tempo pré-determinado
- Quando o tempo acaba, o processo é interrompido.

## ■ Não-preemptivo:

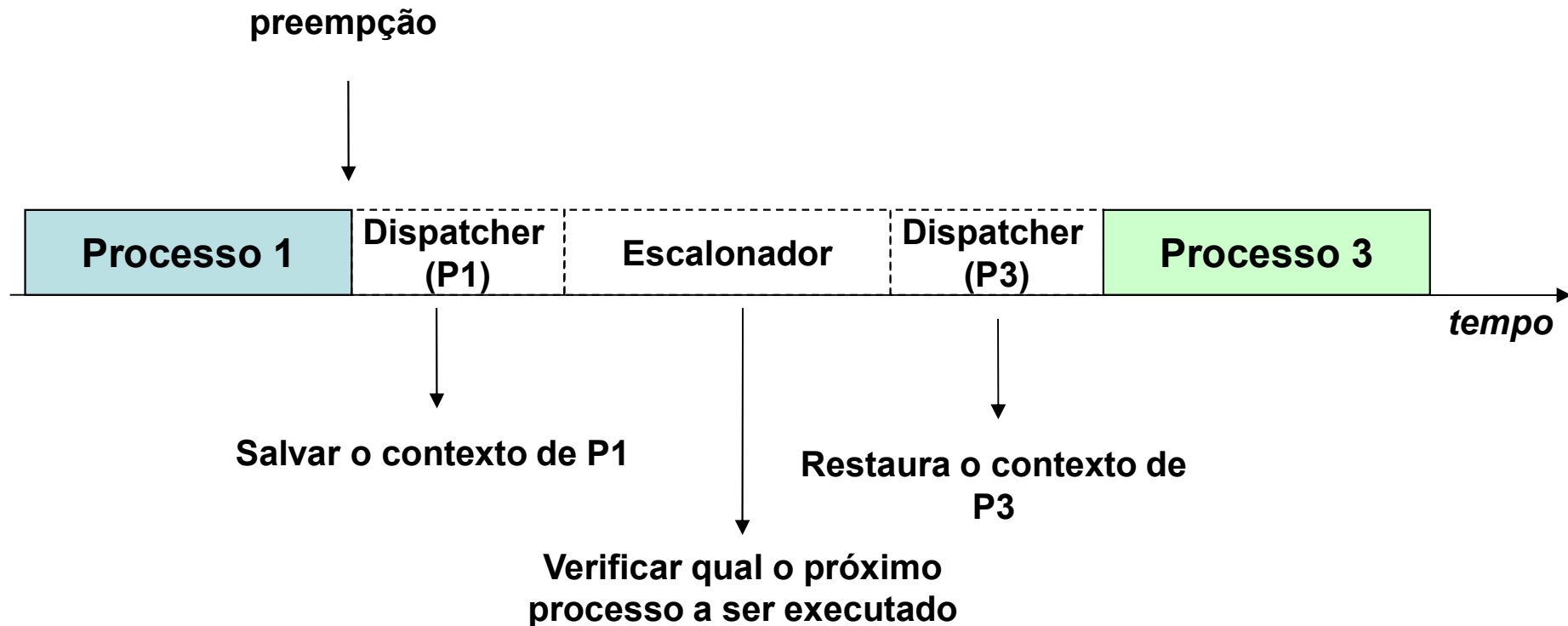
- Processo fica em execução até que:
  - Termine
  - Libere a CPU VOLUNTARIAMENTE
  - Seja bloqueado por falta de recurso

# O que afeta a performance de um algoritmo de escalonamento?

---

- Cada processo possui informações que permitem definir precisamente seu estado.
  - Tais informações definem o **contexto** do processo
- Troca de Contexto
  - Mecanismo que permite ao escalonador interromper uma tarefa, e executá-la posteriormente, sem corromper seu estado.
  - Separação do escalonamento
    - Escalonamento = Política + Mecanismo

# Ilustração da troca de contexto



# Qual o objetivo do escalonamento?

---

- DEPENDE do tipo de sistema operacional
  - Lote:
    - Não possui usuários aguardando → pode ser preemptivo ou não
    - Não possui muita troca de contexto
    - OBJETIVOS:
      - melhorar o throughput (vazão)
      - melhorar o turnaround (tempo entre submissão e finalização)
      - manter a CPU ocupada

# Qual o objetivo do escalonamento?

---

## – Propósito Geral:

- Possuem usuários interagindo
- Precisam ser preemptivos
- OBJETIVOS
  - melhorar o tempo médio de resposta
  - atender as expectativas dos usuários

## – Tempo real:

- Em geral são preemptivos
- OBJETIVO:
  - cumprir requisitos lógicos
  - cumprir requisitos temporais



# Qual o objetivo do escalonamento?

---

- Independente do tipo de sistema operacional, TODOS os algoritmos de escalonamento precisam atender a alguns critérios:
  - Justiça (fairness)
  - Aplicação da política de escalonamento
  - Equilíbrio (balance) entre as partes do sistema

# Escalonamento para sistemas em lote

---

## ■ FCFS (ou FIFO)

- Primeiro processo da fila de pronto é o escolhido para executar.
- Não-preemptivo
- Fácil de entender
- Fácil de programar
- “Justo”
- Processos de baixo custo de execução podem esperar muito tempo para ser executado

# Escalonamento para sistemas em lote

## ■ FCFS (ou FIFO)

– Fazer o escalonamento para os seguintes processos:

PROCESSO	CUSTO DE EXECUÇÃO	INSTANTE DE CHEGADA
A	12	10
B	8	5
C	15	0
D	5	7

Tempo de resposta: tempo final – instante de chegada

$$TR(A) = 40 - 10 = 30$$

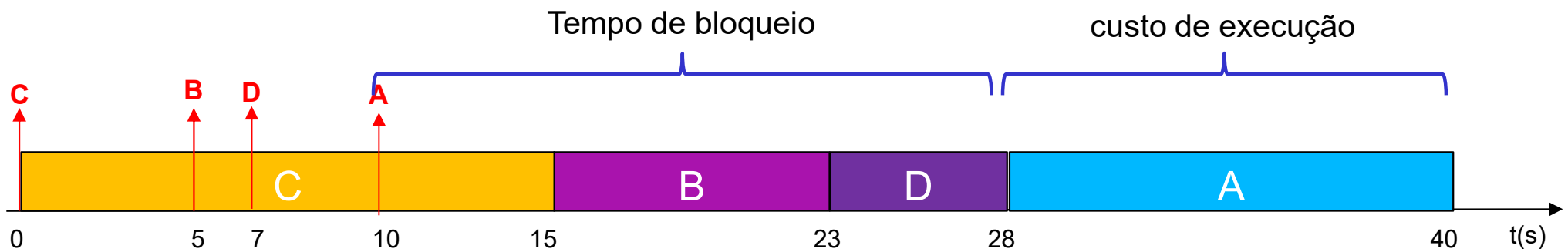
$$TR(B) = 23 - 5 = 18$$

$$TR(C) = 15 - 0 = 15$$

$$TR(D) = 28 - 7 = 21$$

$$TR(\text{médio}): 21,5$$

Tempo de resposta: custo do processo + tempo de bloqueio



# Escalonamento para sistemas em lote

---

- Menor Job Primeiro (MJP)
  - O *job* de menor custo de execução executa primeiro.
  - Não-preemptivo
  - Fácil de entender
  - Fácil de programar
  - “Justo”
  - Para ser adequado requer que todos os jobs estejam disponíveis simultaneamente

# Escalonamento para sistemas em lote

## ■ Menor Job Primeiro

- Fazer o escalonamento para os seguintes processos

PROCESSO	CUSTO DE EXECUÇÃO	INSTANTE DE CHEGADA
A	12	0
B	8	0
C	15	0
D	5	0

Tempo de resposta: tempo final – instante de chegada

$$TR(A) = 25 - 0 = 25$$

$$TR(B) = 13 - 0 = 13$$

$$TR(C) = 40 - 0 = 40$$

$$TR(D) = 5 - 0 = 5$$

$$TR(\text{médio}): 20,75$$

Tempo de resposta: custo do processo + tempo de bloqueio

